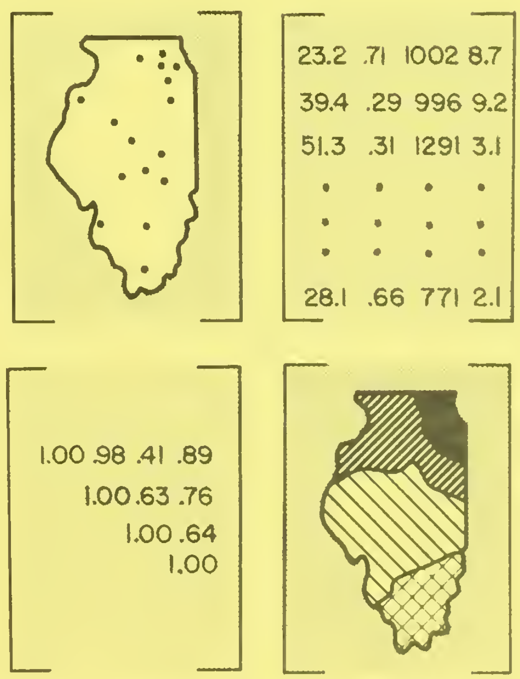
OCCASIONAL PUBLICATIONS OF THE DEPARTMENT OF GEOGRAPHY

MATRIX AND GRAPHIC SOLUTIONS TO THE TRAVELING SALESMAN PROBLEM

by

ROSS MULLNER



NOVEMBER 1972
PAPER NUMBER 4

JUDITH I. PARKHURST and GARY O. ANDERSON, editors

GEOGRAPHY GRADUATE STUDENT ASSOCIATION
UNIVERSITY OF ILLINOIS at URBANA - CHAMPAIGN

MATRIX AND GRAPHIC SOLUTIONS TO THE TRAVELING SALESMAN PROBLEM

Ross Mullner

ABSTRACT

Large companies employing fleets of trucks minimize their trucking routes by utilizing optimization techniques. These companies can economically afford to hire operations research analyzers and computer programmers to find their optimal routes. Conversely, small companies who lack the use of a computer attempt to find their optimal routes by trial and error. This paper explores alternatives to the trial and error method by demonstrating various traveling salesman algorithms that can be utilized without the use of a computer. Specifically the objectives of this study are: (1) to derive optimal routes for the company under study, and (2) to compare solutions of various traveling salesman algorithms in order to recommend the best solution for optimization that could be economically employed by this and other small companies. Three matrix or graphic solutions were compared, the Heinritz-Hsiao algorithm, the Cascade algorithm, and the Lockset method of sequential programming. After comparison, the Lockset Method was recommended to the company.

INTRODUCTION

A classical optimization problem is that of the traveling salesman. Simply stated, it involves finding an optimal route between a series of locations or stops, under the condition that each stop is visited once and only once and a return is made to the point of origin. An optimal route may be defined in several ways such as the route having the smallest distance traveled, the least travel cost, or the smallest travel time.

The traveling salesman problem can be divided into two types, symmetric and asymmetric. A problem is considered symmetric if the routes between any two points are the same (i.e., the same distance, travel cost, or travel time) regardless of the direction traversed. For an asymmetric problem routes vary between any two given points, such as one way streets where routes cannot be retraced.

Small problems involving only a few stops are typically solved by listing all possible routes and selecting the shortest route among the alternatives. This method is known as direct search. However, for larger problems the number of possible routes increases enormously and other methods must be employed. For example, a symmetric problem with twelve stops has nearly 240 million different route possibilities (the total number of possible routes not counting the reverse of such routes is equal to 1/2 N!). Adding one more stop to the above problem increases the total possible routes to nearly three billion (Schruben and Clifton, 1968, p. 855).

Since the traveling salesman problem was first posed by Whitney during a seminar in 1934 (Flood, 1956, p. 61), there have been many algorithms developed which derive near optimal solutions. Some algorithms are matrix and graphic solutions while others are mathematically more complex. The majority of current algorithms are of the mathematically complex type and require the use of a digital computer. Examples of the above include: dynamic programming, integer programming, linear programming, branch-and-bound, tour-to-tour approximations and the Gilmore-Gomory method. In a review of various computer derived solutions Billmore and Nemhauser state:

> If the authors were faced with the problem of finding a solution to a particular traveling salesman problem we would use dynamic programming for problems with 13 cities or less, Shapiro's branch-and-bound algorithms for larger problems (up to about 70-100 cities for asymmetric problems and up to about 40 cities for symmetric problems) and Shen Lin's '3-opt'algorithm for problems that cannot be handled by Shapiro's algorithm. We recommend dynamic programming over branch-and-bound for smaller problems, although the expected computer time might be greater, we are assured that the maximum time is very small (Bellmore and Nemhauser, 1968, p. 556).

Hence, no one method is applicable for solving all traveling salesman problems. "The problem is not that of knowing how to find the solution. The problem is that of knowing how to find the solution easily" (Garrison, 1960, p. 358). But, as of yet there is no one, simple, efficient, mathematical procedure (Maffei, 1965, p. 16).

PROBLEM

Large companies employing fleets of trucks are quick to learn and utilize optimization techniques. These firms hire teams of operations research analyzers and computer programmers to find their optimal or near-optimal routes. By doing so an enterprise can save time, fuel, wear on equipment, and utilize its manpower to a greater degree. Smaller companies, however, rarely can afford the services of such people and generally lack the use of a computer. How then does a small firm attempt to find its optimal routes? "The typical method in use today is one of trial and error, and generally consists of looking at a map, picking out routes consistent with available carrier capacities, and then by trial and error attempting to find shorter routes" (Cochran, 1967, p. 2).

This paper explores alternatives to the trial and error method by demonstrating and comparing various traveling salesman algorithms applicable to one small propane gas distributing company in suburban Chicago. Specifically, the objectives of this study are: (1) to derive optimal routes that minimize mileage on the trucking routes of a small company; and (2) to compare the solutions of various traveling salesman algorithms to the optimal routes in order to recommend the best solution for optimization that could be economically employed by small companies.

The actual algorithms chosen for comparison were subject to two constraints: (1) the company under study did not own or have access to a digital computer, therefore all solutions would have to be solved by hand computation; and (2) the particular algorithm recommended for use would have to be as simple and efficient as possible in order to limit computation time and cost.

In searching for algorithms which would satisfy these two constraints, three matrix or graphic solutions were accepted, while seven other methods, including those six previously discussed were rejected. The six rejected methods included: dynamic programming, integer programming, linear programming, branch-and-bound, tour-to-tour approximation, and the Gilmore-Gomory method. These algorithms, although easily solved by use of a computer, were found to require excessive computation time by hand. For example, for a five stop problem employing zero-one programming, a special case of integer programming, there were twenty variables and twenty constraints. A six stop problem increases the number of variables to thirty and the number of constraints to 67 (Plane and McMillan, 1971). The seventh method to be rejected was a graphic solution formulated by Barachet (1957). This method, after being applied to various sample problems, was found to be both difficult to employ and time consuming.

The three solutions chosen for comparison include: (1) the Heinritz-Hsiao method (Heinritz and Hsiao, 1969), (2) the Cascade algorithm (Haggett and Chorley, 1969), and (3) the Lockset method of sequential programming (Schruben and Clifton, 1968). Each of the algorithms will be discussed below, after a brief discussion of the data collection method.

METHODOLOGY

The data collection for this study consisted of distributing a set of large scale road maps of the city of Chicago and its suburbs to each of the company's truckdrivers. On the first series of maps each driver was asked to trace out his main or original routes and to indicate every stop made. On the next series of maps each driver was asked to plot the route he would take from the company directly to each stop. Lastly, each driver was asked to trace out a route from each stop to every other stop. An example of one of the routes is illustrated in Figure 1 (p. 7).

Distance measurements were taken directly from the maps and put into matrix form. The distance matrices for the main routes are shown in Tables 1 - 4.

TABLE 1 -- DISTANCE IN MILES FOR ROUTE 1

Stops	1	2	3	4	5	6
1						
2	11.37					
3	11.65	.70				
4	10.13	1.70	1.66			
5	9.78	2.12	2.46	.40		
6	10.31	1.62	1.90	.41	.57	

TABLE 2 -- DISTANCE IN MILES FOR ROUTE 2

Stops	1	2	3	4	5	6	7	8
1								
2	5.35							
3	4.58	3.14						
4	5.31	4.10	1.11					
5	5.41	4.20	1.21	.10				
6	5.23	4.00	1.02	1.19	.29			
7	5.35	4.13	1.15	.32	.42	.13		
8	3.20	3.29	2.45	2.18	2.26	2.11	2.25	

TABLE 3 -- DISTANCE IN MILES FOR ROUTE 3

Stops	1	2	3	4	5	6	7	8
1								
2	6.80							
3	7.85	1.30						
4	8.75	2.31	.09					
5	11.93	6.78	3.90	3.50				
6	16.40	7.97	6.75	6.36	5.05			
7	14.50	9.01	8.40	9.16	4.82	10.96		
8	5.04	8.62	7.58	7.28	10.92	10.45	15.50	

TABLE 4 -- DISTANCE IN MILES FOR ROUTE 4

Stops	1	2	3	4	5	6	7	8	9	10	11	12
1												
2	.45											
3	3.30	3.11										
4	6.78	6.62	4.10									
5	10.30	10.15	7.64	5.82								
6	5.89	5.58	8.60	8.40	9.15							
7	6.12	5.89	8.90	8.71	9.50	1.50						
8	7.71	7.49	9.00	8.80	9.58	1.60	.42					
9	4.00	3.71	6.70	6.57	9.20	5.87	4.61	4.18				
10	3.72	3.43	6.56	6.29	8.95	5.60	4.68	4.24	.25			
11	3.04	2.79	5.79	5.67	8.30	4.95	5.74	5.29	1.31	1.05		
12	2.62	2.34	5.34	6.30	8.95	5.60	6.87	5.92	1.98	1.67	1.05	

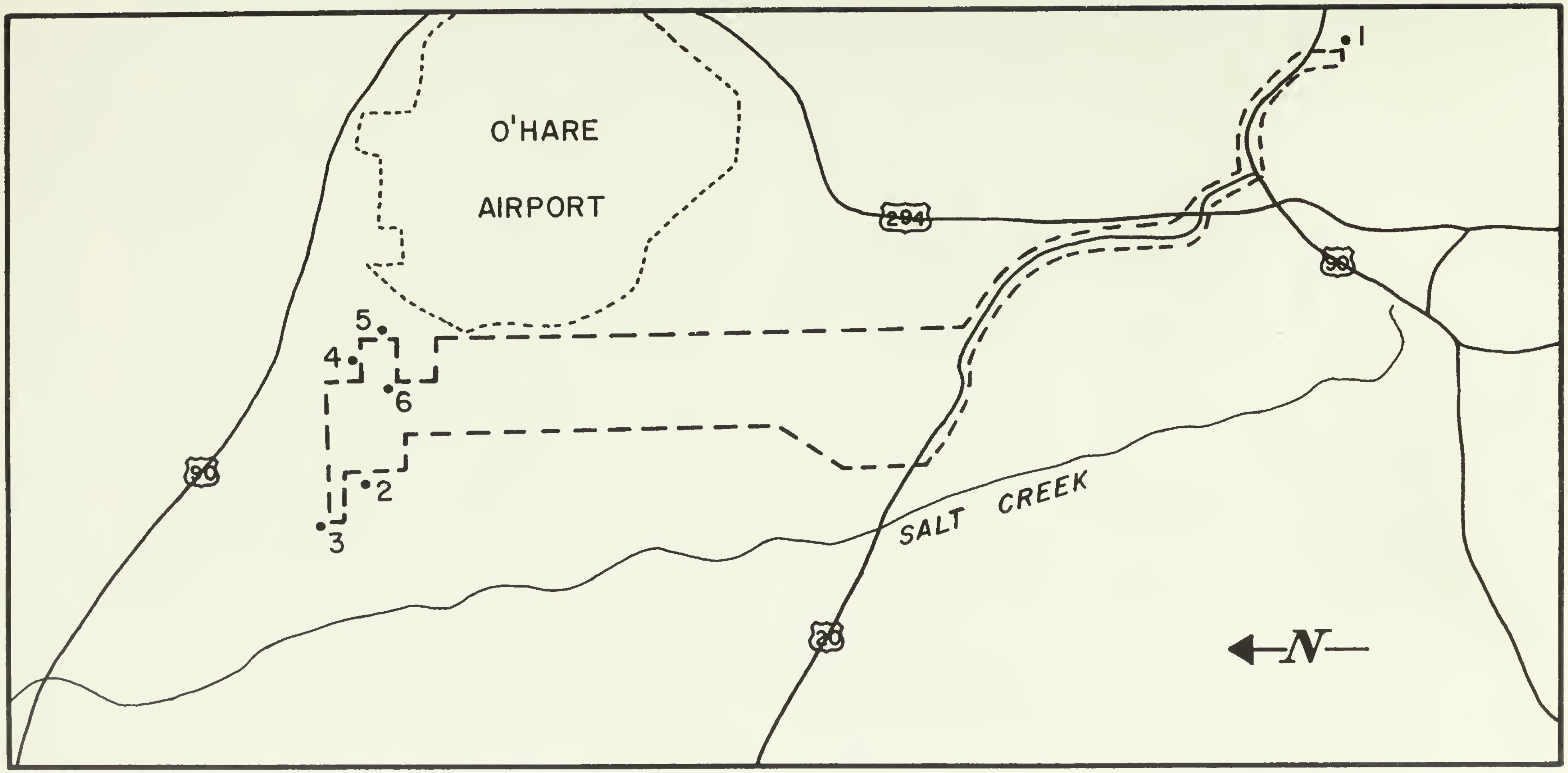


FIGURE 1 -- AN EXAMPLE OF ONE OF THE ROUTES (ROUTE 1)

After collecting the data, a small computer program (see Appendix) was written to find the optimal route for three of the four main routes. By employing direct search for the three routes, the optimal solutions could be compared to those solutions generated by the three algorithms, thus serving as one means of evaluation. The optimal solution for the fourth route could not be found for the number of route possibilities exceeds the storage capacity of the largest computer. The optimal and original routes are listed in Table 5.

TABLE 5 -- COMPARISON OF OPTIMAL AND ORIGINAL ROUTES

	Optimal Routes	Original Routes	Distance Saved	Percent Saved
1	24.46	25.01	.55	2.19
2	15.52	15.57	.05	.32
3	42.71	48.24	5.53	11.46
4	-	33.70	-	-

The first algorithm to be discussed was developed by Heinritz and Hsiao (Heinritz and Hsiao, 1969). These authors, attempting to find the minimum cost route for the distribution of centrally processed library material, developed a solution to the traveling salesman problem. The authors state their algorithm is accurate, can achieve a near-optimal solution, and requires no mathematical background.

An example of the solution as employed by Heinritz and Hsiao is illustrated by a main library (A) which must distribute material to eight branch libraries (B-I). The first step in the procedure is to calculate the cost of transporting the material from the main library to each branch library. Next, the cost of transporting the material from each branch to every other branch library is computed. These costs are entered into a matrix (Table 6)

TABLE 6 -- HEINRITZ-HSIAO ALGORITHM PROCEDURE

		TO								
		A	B	C	D	E	F	G	H	I
	A	-	4.8	2.0	1.6	2.8	3.3	4.9	2.3	0.8
	B	4.8	-	3.6	5.6	6.8	1.9	9.6	2.8	5.6
	C	2.0	3.6	-	2.1	4.3	2.0	6.3	2.1	2.8
	D	1.6	5.6	2.1	-	4.0	4.1	4.3	3.5	1.9
FROM	E	2.8	6.8	4.3	4.0	-	6.2	4.3	4.1	2.2
	F	3.3	1.0	2.0	4.1	6.2	-	8.3	2.3	4.5
	G	4.9	9.6	6.3	4.3	4.3	8.3	-	7.1	4.2
	H	2.3	2.8	2.1	3.5	4.1	2.3	7.1	-	2.9
	I	0.8	5.6	2.8	1.9	2.2	4.5	4.2	2.9	-

Once the matrix is derived, the row representing the starting point is entered and the lowest value in that row is circled. The Table 6 starting point is row A and the value circled is 0.8 in column I. Since the A to I portion of the route has been established, the second move is from I. Therefore, row I is entered and the lowest value in that row is circled. However, since each stop is to be visited only once, no return to A is permitted except to terminate the route. Thus, the value circled is 1.9 in column D. Entering row D and ignoring previous stops, the lowest value is again circled and so on. The last circle will establish a return to the starting point and will terminate the route. For the above example, the near-optimum route is A-I-D-C-F-B-H-E-G-A.

In applying this procedure to the four trucking routes under study, two solutions were greater than the original routes, while the other two solutions were found to be less than the original routes. The original routes and those derived by the algorithm are given in Table 7.

TABLE 7 -- COMPARISON OF HEINRITZ-HSIAO ALGORITHM AND ORIGINAL ROUTES

Original Routes	Heinritz-Hsiao Routes
25.01	24.56
15.57	15.56
48.24	52.67
33.70	41.68

The next algorithm to be examined is the Cascade method. This method which has been employed by Murchland (1965) in finding elementary paths in a complete directed graph but it can also be used in solving traveling salesman problems. The Cascade method uses a matrix, the elements of which represent either cost, distance, or time between stops. Blanks are used to indicate unknown values which cannot be estimated. These blanks are assumed to be larger than the total sum of all elements in the matrix. Each element along the main diagonal serves in turn as a pivot. All combinations of one element from the pivot row and pivot column are summed. If any of these sums are smaller than the value at the row and column intersect, the sum replaces the value at that intersect. When the entire procedure is completed for all pivot points, the matrix indicates the optimal route (Haggett and Chorley, 1969, p. 201).

For example, a hypothetical four by four matrix, consisting of known and/or estimated values, is constructed (Table 8A). Each blank

TABLE 8 -- PROCEDURE EMPLOYED BY THE CASCADE ALGORITHM

Matrix A

From	To: A	B	C	D
A	0	2	–	–
B	1	0	3	5
C	6	4	0	–
D	8	–	–	0

Matrix B

From	To: A	B	C	D
A	0	2	–	–
B	1	0	3	5
C	6	4	0	–
D	8	10	–	0

Matrix C

From	To: A	B	C	D
A	0	2	5	7
B	1	0	3	5
C	5	4	0	9
D	8	10	13	0

Matrix D

From	To: A	B	C	D
A	0	2	5	7
B	1	0	3	5
C	5	4	0	9
D	8	10	13	0

cell is indicated here by a dash. A_{11} is the initial pivot point. At step one every possible pair of cells in the pivot row and pivot column is summed. When a sum such as $A_{12} + A_{41}$ is smaller than the value of the intersect (A_{42}), the sum, which is ten, replaces the original value. Table 8B, C, and D completes the matrix for each successive pivot point. The minimum route, assuming point A is the origin, is found by entering row 1 of the final matrix (Table 8D) and moving to the lowest value in that row. In matrix D this value is 2 in column 2. Row 2 is entered to find the third minimum route. Column 3 has the lower remaining number, and so on. Thus, the minimum route is A-B-C-D.

One advantage of this method is that it is not necessary to enter all values into the matrix in order to derive a solution. Thus, if the value between two points is difficult to obtain and cannot be estimated, it can be ignored.

After applying the Cascade algorithm to the four trucking routes, the results were found to be similar to those derived by the Heinritz-Hsiao method. The solutions from the two methods were the same with the exception of one route. Similarly, two of these solutions were greater than the original routes, while two solutions were less than the original routes. The results of the Cascade algorithm as compared to the optimal, original, and Heinritz-Hsiao routes are shown in Table 9.

TABLE 9 -- COMPARISON OF OPTIMAL, ORIGINAL AND ROUTES GENERATED BY THE TWO ALGORITHMS

Optimal Routes	Original Routes	Heinritz-Hsiao Routes	Cascade Routes
24.46	25.01	24.56	24.56
15.52	15.57	15.56	15.56
42.71	48.24	52.67	48.56
-	33.70	41.68	41.68

The final algorithm under consideration is the Lockset method of sequential programming. This method, which has been applied to routing delivery and pickup trucks by Schruben and Clifton (1968), assumes a maximum initial route where each stop is connected directly to the origin (Figure 2a). This initial route is then modified by joining stops through a series of successive aggregations. An example of the method applied to a small problem is given below.

Suppose there is a company which supplies a product to four customers. The minimal route, using the Lockset method, is found by constructing a distance matrix (Table 10). A list of all possible pairs

TABLE 10 -- DISTANCE MATRIX FOR THE LOCKSET PROBLEM

	Plant	Customer 1	Customer 2	Customer 3	Customer 4
Plant					
Customer 1	34				
Customer 2	47	17			
Customer 3	67	34	26		
Customer 4	48	23	34	31	

(after Schruben and Clifton, 1968, p. 855)

of stops not involving the plant (or origin) is compiled. From this list, distance-saved coefficients (DSC) for each pair of stops are calculated using the following equation

$$P_oP_i + P_oP_j - P_iP_j = DSC$$

where
P_o is the origin
P_i is the point i
P_j is the point j
P_oP_i represents distance between P_o and P_i
P_oP_j represents distance between P_o and P_j
P_iP_j represents distance between P_i and P_j

The pairings and the DSC for the problem are given in Table 11.

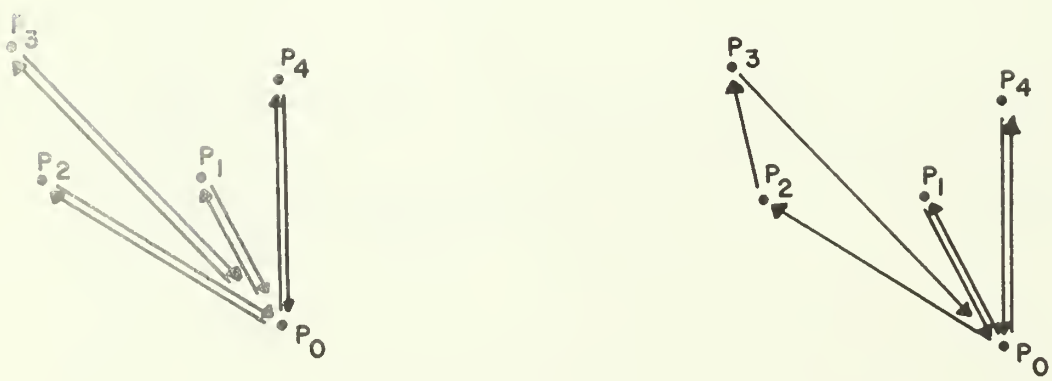
TABLE 11 -- PAIRINGS AND DISTANCE-SAVED COEFFICIENTS

Pairing		Distance-saved coefficient			
P_i	P_j	P_oP_i	P_oP_j	P_iP_j	DSC
P2 with	P1	47	34	17	64
P3 with	P1	67	34	34	67
P3 with	P2	67	47	26	88
P4 with	P1	48	34	23	59
P4 with	P2	48	47	34	61
P4 with	P3	48	67	31	84

The pair of stops with the largest DSC are joined on the same route, if possible. For this problem P2 and P3 are combined resulting in the first aggregation P_o-P_2-P_3-P_o. Each revised route is tested on the basis of the following criteria: (1) each stop must have at least one leg connected with the origin, and (2) each stop must have previously been on a different route. Therefore, joining P_2P_3 is accepted for both conditions are met. This leg will be retained throughout subsequent aggregations for it is "locked in" the route set. The initial route and the first aggregation are illustrated in Figure 2. Following the same method, the next largest DSC (P_3P_4) is tested and combined forming a revised route. Again both criteria are met. The second aggregation P_o-P_2-P_3-P_4-P_o is shown in Figure 3a.

FIGURE 2 -- ASSUMED MAXIMUM INITIAL ROUTES

(a) (b)



(after Schruben and Clifton, 1968)

FIGURE 3 -- ASSUMED MAXIMUM INITIAL ROUTE

(a) (b)

(after Schruben and Clifton, 1968)

This procedure is continued until all of the DSC pairs have been tested. If any DSC does not meet one of these conditions, it is rejected and the next lower DSC is tested, and so on. When all DSC pairs have been tested the optimal route has been found. For this problem the optimal route is P_o-P_1-P_2-P_3-P_4-P_o and is shown in Figure 3b.

Application of the Lockset method to the four trucking routes under study resulted in three solutions which were less than the original routes and one solution which was greater than the original route. An example of one of the Lockset routes and the results of this method as compared to the optimal and original routes and those of the other two algorithms are given in Figure 4 (p. 17) and Table 12.

TABLE 12 -- COMPARISON OF OPTIMAL, ORIGINAL AND ROUTES GENERATED BY THE THREE ALGORITHMS

Optimal Routes	Original Routes	Heinritz-Hsiao Routes	Cascade Routes	Lockset Routes
24.46	25.01	24.56	24.56	24.46
15.52	15.57	15.56	15.56	15.53
42.71	48.24	52.67	48.56	45.78
-	33.70	41.68	41.68	34.47

CONCLUSION

After comparing the results and the computational efficiency of the three algorithms, the Heinritz-Hsiao and Cascade methods were rejected as possible solutions for the company, while the Lockset method was accepted. This method, although requiring slightly more computation than the other algorithms, was found to outweigh this minor disadvantage by its greater accuracy.

The Lockset method was chosen because it was the only algorithm to find the optimal solution for one of the four routes (24.46) (Table 12). Second, this was the only method to generate three solutions which were less than the original routes. Third, the solution which was greater than the original route was only slightly larger than the original route as compared to the results of the other two algorithms. Last, the original route and solution crossed themselves thus indicating the original

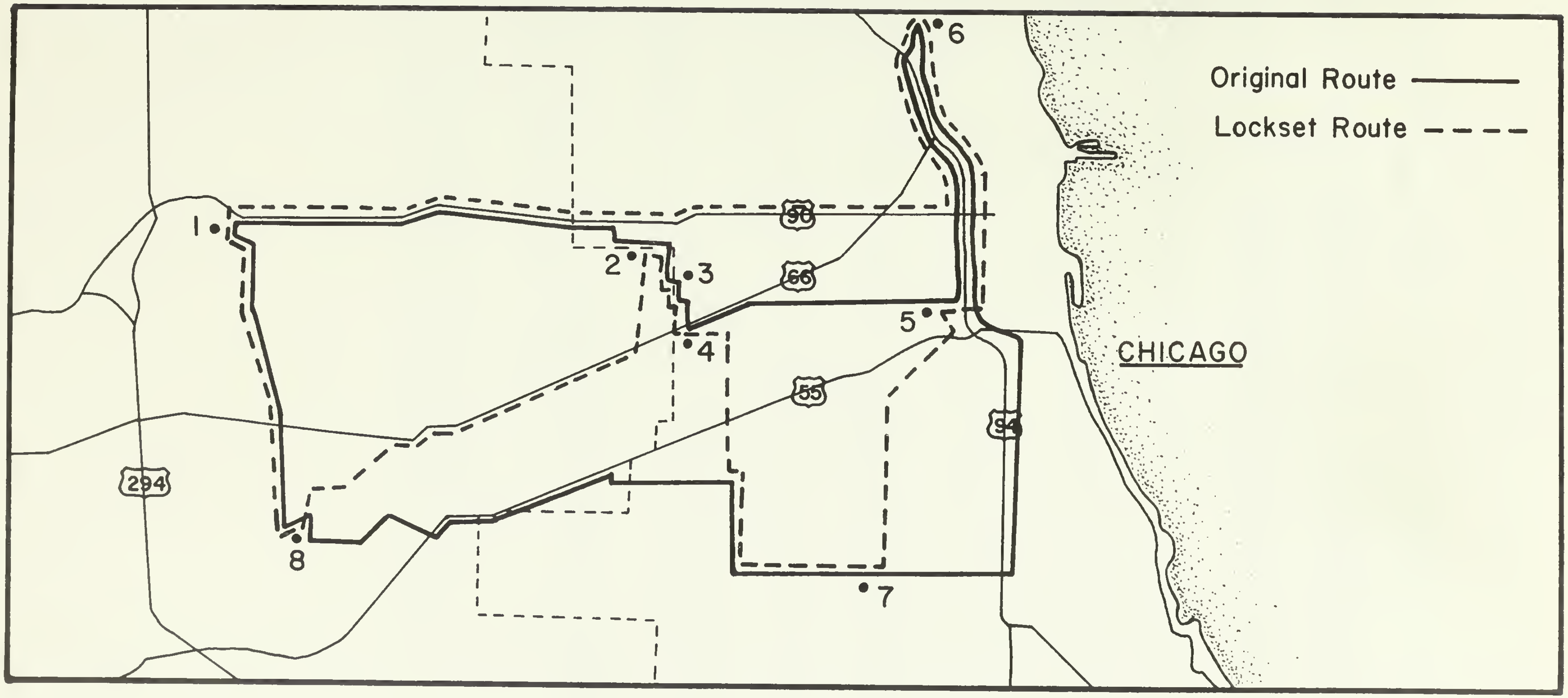


FIGURE 4 -- ROUTE 3 AND THE LOCKSET ROUTE

route was not the minimal possible route (Barachet, 1957). Further, it was found that if those links which crossed themselves were modified into straight routes between stops in the original Lockset solution, the route generated would be less than the original route. For the route indicated in Figure 4 this modification resulted in a solution which was 33.61 miles long as compared to the original route's 33.70 miles. The Lockset method and its modifications generated four solutions which were less than the original routes. Therefore, it was recommended to the company.

REFERENCES

Barachet, L. L., 1957. Graphic Solution of the Traveling Salesman Problem, Operations Research, vol. 5, p. 841-845.

Bellmore, M. and G. L. Nemhauser, 1968. The Traveling-Salesman Problem: A Survey, Operations Research, vol. 16, p. 538-558.

Cochran, Harold, 1967. Optimization of a Carrier Routing Problem, Unpublished M.S. Thesis, Kansas State University, 122 p.

Flood, Merrill M., 1956. The Traveling-Salesman Problem, Operations Research, vol. 4, p. 61-75.

Garrison, William L., 1960. The Spatial Structure of the Economy III, Annals of the Association of American Geographers, vol. 50, (no. 3, 1960), p. 357-373.

Haggett, Peter and Richard J. Chorley, 1969. Network Analysis in Geography. London: Edward Arnold, 348 p.

Heinritz, Fred J. and James C. Hsiao, 1969. Optimum Distribution of Centrally Processed Material, Library Resources and Technical Services, vol. 13, no. 2, p. 206-208.

Maffei, Richard B., 1965. Modern Methods for Local Delivery Route Design, Journal of Marketing, vol. 29, p. 13-18.

Murchland, J. D., 1965. A New Method for Finding All Elementary Paths in A Complete Directed Graph, London School of Economics, Transport Networks Theory Group, Report 22, 58 p.

Plane, Donald R. and Claude McMillan, 1971. Discrete Optimization: Integer Programming and Network Analysis for Management Decisions. New York: Prentice-Hall, 251 p.

Schruben, Leonard W. and Ruth E. Clifton, 1968. The Lockset Method of Sequential Programming Applied to Routing Delivery and Pickup Trucks, American Journal of Agricultural Economics, vol. 50, p. 854-867.

APPENDIX

```
      DIMENSION A(8,8)
      N=8
      READ (5,51) ((A(I,J)-1,N),I=1,N)
   51 FORMAT (8F5.2)
      DO 100 I=2,N
      SUM=0.0
      SUM=SUM+A(1,I)
      DO 90 J=2,N
      SUM1=SUM
      IF (J.EQ.I)  GO TO 90
      IF (I.LT.J) SUMI=SUMI+A(I,J)
      IF (J.LT.I) SUMI=SUMI+A(J,I)
      DO 80 K=2,N
      SUM2-SUM1
      IF (K.EQ.I .OR.K.EQ.J)  GO TO 80
      IF (J.LT.K) SUM2=SUM2+A(J,K)
      IF (K,LT,J) SUM2=SUM2+A(K,J)
      DO 70 L=2,N
      SUM3=SUM2
      IF (L.LE.I.OR.L.EQ.K)  GO TO 70
      IF (K.LT.L) SUM3=SUM3+A(K,L)
      IF (L.LT.K) SUM3=SUM3+A(L,K)
      DO 60 M=2,N
      SUM4=SUM3
      IF(M.EQ.I.OR.M.EQ.J.OR.M.EQ.K.OR.M.EQ.L)  GO TO 60
      IF(L.LT.M.) SUM4=SUM4+A(L,M)
      IF(M.LT.L) SUM4-SUM4+A(M,L)
      DO 50 I1=2,N
      SUM5=SUM4
      IF(I1.EQ.I.OR.EQ.J.OR.I1.EQ.K.OR.I1.EQ.L.OR.I1.EQ.M)GO TO 50
      IF(M.LT.I1) SUM5=SUM5+A(M,I1)
      IF(I1.LT.M) SUM5=SUM5+A(I1,M)
      DO 40 12=2,N
      SUM6=SUM5
      IF(I2.EQ.I.OR.I2.EQ.J.OR.I2.EQ.K.OR.I2.EQ.L.OR.I2.EQ.M.OR.I2.EQ.
      1I1  ) GO TO 40
      IF(I1.LT.I2) SUM6=SUM6+A(I1,I2)
      IF(I2.LT.I1) SUM6=SUM6+A(I2,I1)
      SUM6=SUM6+A(1,I2)
      WRITE(6,200) I,J,K,L,M,I1,I2,SUM6
   40 CONTINUE
   50 CONTINUE
   60 CONTINUE
   70 CONTINUE
   80 CONTINUE
   90 CONTINUE
  100 CONTINUE
  200 FORMAT(' 0',7I3,'  0'F10.2)
      STOP
      END
```